

Article

# An Improved Graph Isomorphism Network for Accurate Prediction of Drug–Drug Interactions

Sile Wang <sup>1</sup>, Xiaorui Su <sup>2</sup>, Bawei Zhao <sup>2</sup> , Pengwei Hu <sup>2</sup>, Tao Bai <sup>1,\*</sup> and Lun Hu <sup>2,\*</sup>

<sup>1</sup> College of Computer and Information Engineering, Xinjiang Agricultural University, Urumqi 830052, China; 320213582@xjau.edu.cn

<sup>2</sup> Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China

\* Correspondence: bt@xjau.edu.cn (T.B.); hulun@ms.xjb.ac.cn (L.H.)

**Abstract:** Drug–drug interaction (DDI) prediction is one of the essential tasks in drug development to ensure public health and patient safety. Drug combinations with potentially severe DDIs have been verified to threaten the safety of patients critically, and it is therefore of great significance to develop effective computational algorithms for identifying potential DDIs in clinical trials. By modeling DDIs with a graph structure, recent attempts have been made to solve the prediction problem of DDIs by using advanced graph representation learning techniques. Still, their representational capacity is limited by isomorphic structures that are frequently observed in DDI networks. To address this problem, we propose a novel algorithm called DDIGIN to predict DDIs by incorporating a graph isomorphism network (GIN) such that more discriminative representations of drugs can thus be learned for improved performance. Given a DDI network, DDIGIN first initializes the representations of drugs with Node2Vec according to the topological structure and then optimizes these representations by propagating and aggregating the first-order neighboring information in an injective way. By doing so, more powerful representations can thus be learned for drugs with isomorphic structures. Last, DDIGIN estimates the interaction probability for pairwise drugs by multiplying their representations in an end-to-end manner. Experimental results demonstrate that DDIGIN outperforms several state-of-the-art algorithms on the ogbl-ddi (Acc = 0.8518, AUC = 0.8594, and AUPR = 0.9402) and DDInter datasets (Acc = 0.9763, AUC = 0.9772, and AUPR = 0.9868). In addition, our case study indicates that incorporating GIN enhances the expressive power of drug representations for improved performance of DDI prediction.

**Keywords:** drug–drug interaction network; link prediction; graph isomorphism network; graph representation learning

**MSC:** 92-08



**Citation:** Wang, S.; Su, X.; Zhao, B.; Hu, P.; Bai, T.; Hu, L. An Improved Graph Isomorphism Network for Accurate Prediction of Drug–Drug Interactions. *Mathematics* **2023**, *11*, 3990. <https://doi.org/10.3390/math11183990>

Academic Editor: Ian Morilla

Received: 4 September 2023

Revised: 12 September 2023

Accepted: 18 September 2023

Published: 20 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Drug–drug interactions (DDIs) refer to a condition in which the activity of one drug changes due to the presence of other drugs when more than two drugs are taken simultaneously or consecutively [1,2]. As a common treatment practice, multidrug prescribing is often associated with increased clinical risk, which consequently leads to many adverse drug effects that cause severe injuries to patients and are even responsible for deaths [3,4]. As has been pointed out by [5], about 15% of older adults taking multiple medications are at risk for severe adverse outcomes due to the presence of DDIs. Consequently, DDIs are a major clinical problem in patient safety [6,7], and their existence has become one of the serious threats to public health. As a result, it is important to investigate the effects of DDIs for human health.

Recently, computational algorithms for estimating candidate DDIs have been rapidly developed due to their promising performance, and they can minimize unexpected adverse drug reactions and maximize synergistic benefits when treating disease to some

extent [8]. In addition, network-based computational algorithms approach the DDI prediction problem from a holistic perspective, typically leveraging existing DDIs to construct a comprehensive DDI network [9,10]. Thus, the original DDI prediction problem is formulated as a link prediction problem within the network framework. In order to address it, a majority of network-based algorithms adopt sophisticated graph representation learning models [11], which can be broadly divided into three categories [12], i.e., matrix factorization (MF)-based [13–15], random walk (RW)-based [16], and neural network (NN)-based algorithms [17]. Compared with the other categories, NN-based methods exhibit superior capability in handling graph-structured data and capturing global representations from DDI networks. Therefore, a variety of NN-based prediction algorithms have been developed to identify novel DDIs.

Purkayastha et al. [18] present an effective approach to predict DDIs using rich drug representations by utilizing multiple knowledge sources. In this work, they obtain the representations of all the drug pairs in a given DDI network with LINE [19] and Metapath2vec [20], respectively, and perform the prediction task using a logistic regression classifier. Liu et al. [21] propose a deep attention neural network, DANN-DDI. In this model, SDNE [22] is first utilized to learn drug embeddings. An attention neural network is incorporated to obtain the representations of drug pairs, which are then fed into a deep neural network for DDI prediction. To tackle the challenges posed by structure preservation and sparsity, the DANN-DDI method leverages both first-order and second-order proximities to characterize the local and global network structures. However, there is an obvious disadvantage among these algorithms in that they only learn the embedding of the topology structure, without considering the attributes of nodes, which include chemical structures, targets, enzymes, and so on. As emphasized by [23,24], node attributes are essential for a precise analysis of complicated networks.

Recently, there has been a growing fascination with the utilization of graph neural networks (GNN) [25–28] in DDI prediction. To effectively aggregate the neighboring information, different aggregation strategies have been designed to develop GNN variants. Zitnik et al. [29] propose a novel model, namely, Decagon, which applies a relational GNN for modeling polypharmacy side effects. Asada et al. [30] utilize graph convolutional network (GCN) [31] to encode molecular structures for DDI extraction from the literature. Zhou et al. [32] present a graph distance neural network based on GraphSAGE [33] to predict DDIs, and they comprehensively consider the features of nodes and edges in the graph to better generate the embeddings of drug nodes. Su et al. [34] propose a novel attention-based knowledge graph (KG) representation learning framework, namely, DDKG, which identifies potential DDIs from biomedical KGs in an end-to-end manner. Although these algorithms have achieved strong performance as indicated by their experimental results, their expressive power is further limited by the fact that they fall short of learning distinguishable representations for drugs within isomorphic structures according to the Weisfeiler–Leman (WL) graph isomorphism test [35].

To intuitively explain our motivation, we take a small DDI network associated with the WL problem in Figure 1. This network is automorphic, as the drug nodes  $v_2$  and  $v_4$  are isomorphic to each other. Traditional GNN models, such as GCN and GraphSAGE, attempt to learn similar node embeddings for  $v_2$  and  $v_4$  [36] by following similar aggregation paths, thus yielding close interaction probability scores for the drug pairs, i.e.,  $(v_1, v_2)$  and  $(v_1, v_4)$ . However, in the context of the entire DDI network,  $v_2$  and  $v_4$  are actually placed at different positions away from  $v_1$ . More specifically,  $v_2$  and  $v_4$  are located at three and four hops away from  $v_1$ , respectively. In this regard, it is more reasonable to handle isomorphic structures in a way that maps different node neighborhoods to different embeddings. As a result, the representations learned for drugs are more discriminative even for those within isomorphic structures, thus improving the performance of DDI prediction.

Hence, we propose a novel prediction algorithm, namely, DDIGIN, to complete the DDI prediction task by using a graph isomorphism network (GIN) [37]. In particular, given a DDI network, we first construct a sparse adjacency matrix, and then apply the

Node2Vec [38] embedding approach to initialize drug representations. When propagating the first-order neighboring information of drug nodes, DDIGIN recursively gathers such information in an injective manner so as to accurately learn the global representations of drug nodes by distinguishing isomorphic structures. Finally, DDIGIN calculates the interaction probability for a query pair of drugs by simply multiplying their respective representations. Extensive experiments are performed on two practical datasets of varying sizes to evaluate the effectiveness of DDIGIN. The results indicate that DDIGIN achieves superior performance over several state-of-the-art DDI prediction algorithms.

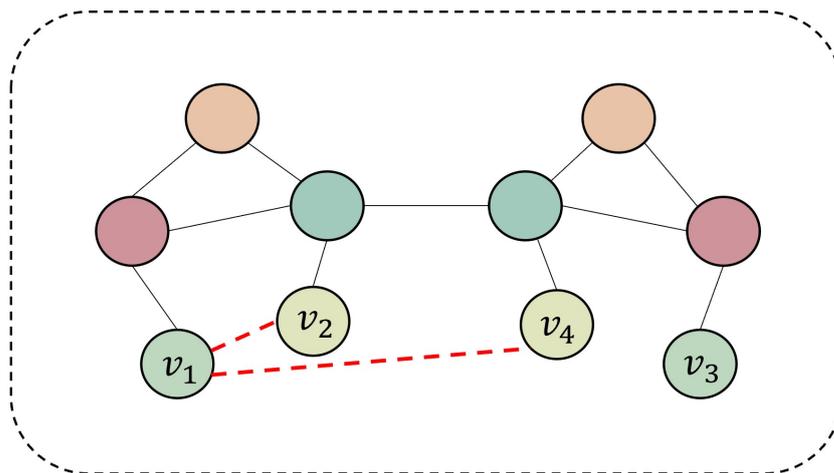


Figure 1. A small DDI network with isomorphic structures.

2. Methods

The overall pipeline of DDIGIN is illustrated in Figure 2. Specifically, DDIGIN consists of three steps: (i) drug embedding initialization, where Node2Vec is applied to initialize drug representations according to the topological structure of DDI network; (ii) drug representation learning, which aims to learn the global representations of drugs by aggregating neighboring information in an injective manner; (iii) DDI prediction, which estimates the interaction probability for drug pairs.

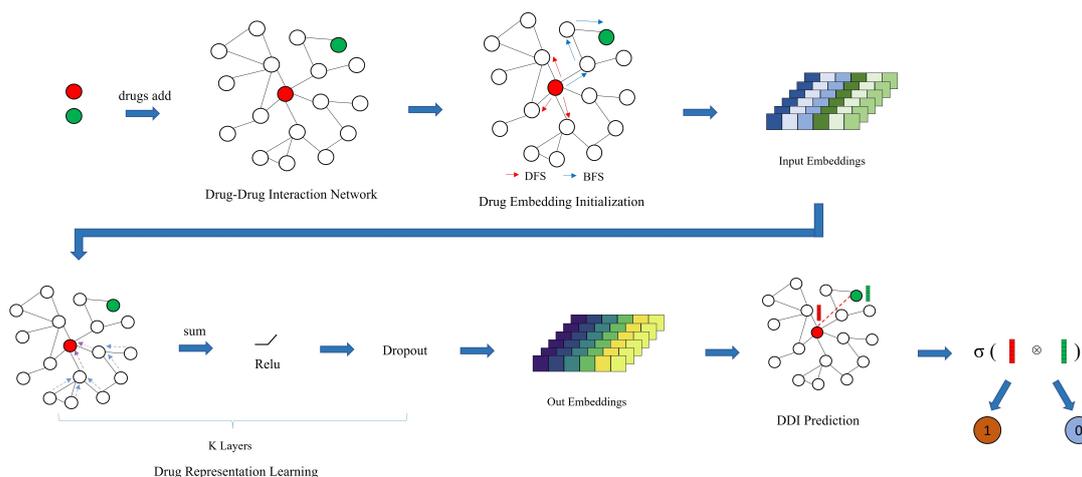


Figure 2. An illustration of the overall pipeline of DDIGIN.

2.1. Notations and Problem Formulation

This section begins by presenting the mathematical notations utilized in our study, followed by formulating the DDI prediction problem employing these notations.

Suppose  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is a graph with the node set  $\mathcal{V}$  and the edge set  $\mathcal{E}$ . Throughout this paper, we denote sets with calligraphic scripts (e.g.,  $\mathcal{G}, \mathcal{N}$ ), vectors with lowercase boldface

letters (e.g.,  $h \in R^d$ ), scalars and drugs with lower characters (e.g.,  $l$  for the number of layers of DDIGIN and  $v_d \in \mathcal{V}$  for drug nodes), and functions with mathematical formal scripts (e.g.,  $\mathcal{F}, \mathcal{L}$ ).

The goal of DDIGIN is to determine whether two arbitrary drugs, i.e.,  $v_i$  and  $v_j$  in  $\mathcal{G}$ , will interact or not. To achieve this, the prediction task is formulated as learning a scoring function  $\mathcal{F}$  for estimating  $\hat{p}_{(i,j)}$ , which is the probability of  $v_i$  interacting with  $v_j$ , expressed as:

$$\hat{p}_{(i,j)} = \mathcal{F}(i, j | \mathcal{G}, \Theta) \tag{1}$$

where  $\Theta$  denotes a set of trainable parameters involved in  $\mathcal{F}$ .

## 2.2. Details of DDIGIN

### 2.2.1. Drug Embedding Initialization

For network-based prediction problems, feature embeddings should be constructed for nodes and edges to form their real-valued representations [39]. DDIGIN first initializes drug embeddings by Node2Vec [38], which uses a biased random walk parameterized by  $p$  and  $q$ . In particular,  $p$  controls the probability of returning to the previous node, and  $q$  controls whether to move outward (in a depth-first-search-like manner) versus inward (in a breadth-first-search-like manner) when encoding both local and global neighbor network characteristics.

- Random walks: Given a drug node  $v_d$ , we generate a sequence  $S$  through a random walk. By denoting  $S_k$  as the  $k$ th drug node in the walk, we take  $S_0 = d$  as the starting point, and  $S_k$  is generated with the following distribution:

$$P(S_k = i | S_{k-1} = j) = \begin{cases} \frac{\Pi_{(i,j)}}{z}, & \text{if } (i, j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where  $\Pi_{(i,j)}$  represents the unnormalized transition probability between drugs  $v_i$  and  $v_j$ , while  $z$  denotes the normalizing constant.

- Search bias: Given a 2nd order random walk with  $p$  and  $q$  guiding the walk, we set  $\Pi_{(i,j)} = \alpha_{pq}(t, i) \cdot w_{(i,j)}$  to evaluate the transition probability between drug nodes, and its value is computed as:

$$\alpha_{pq}(t, i) = \begin{cases} \frac{1}{p}, & \text{if } d_{ti} = 0 \\ 1, & \text{if } d_{ti} = 1 \\ \frac{1}{q}, & \text{if } d_{ti} = 2 \end{cases} \tag{3}$$

where  $d_{ti}$  denotes the shortest path distance between drugs  $v_t$  and  $v_i$ , and  $v_t$  is the previous node of  $v_j$ .

Starting from each drug, we simulate random walks of fixed length in order to compute transition probabilities, and then uses a stochastic gradient descent to optimize drug embeddings so that drugs with similar neighboring nodes in the DDI network are also similar in the feature space.

### 2.2.2. Drug Representation Learning

Regarding drug representation learning, DDIGIN implements it with two components: (i) information propagation and (ii) information aggregation.

**Information propagation.** For a given arbitrary drug node  $v_i$ , the amount of its first-order neighboring information is estimated by the linear combination of node representations defined as:

$$h_{\mathcal{N}_i} = \sum_{j \in \mathcal{N}_i} h_j \tag{4}$$

where  $h_j$  denotes the embedding of  $v_j$ .

**Information aggregation.** DDIGIN introduces a learnable parameter to adjust its own features after each hop aggregation operation. By integrating the adjusted features and the aggregated neighbor features, DDIGIN applies a multilayer perceptron (MLP) that can fit any rules used to update node embeddings so as to make it injective. Hence, we combine all the  $h_{N_i}$  with the initial embedding of  $v_i$ , denoted as  $h_i$ , to update the global representation of  $v_i$ , according to (5):

$$h_i^{(1)} = MLP^{(1)}((1 + \epsilon^{(1)})h_i^{(0)} + h_{N_i}^{(0)}) \tag{5}$$

where  $h_i^{(1)}$  is the result of the aggregation for  $v_i$  at the first layer, and  $\epsilon$  is a learnable parameter or a fixed scalar.

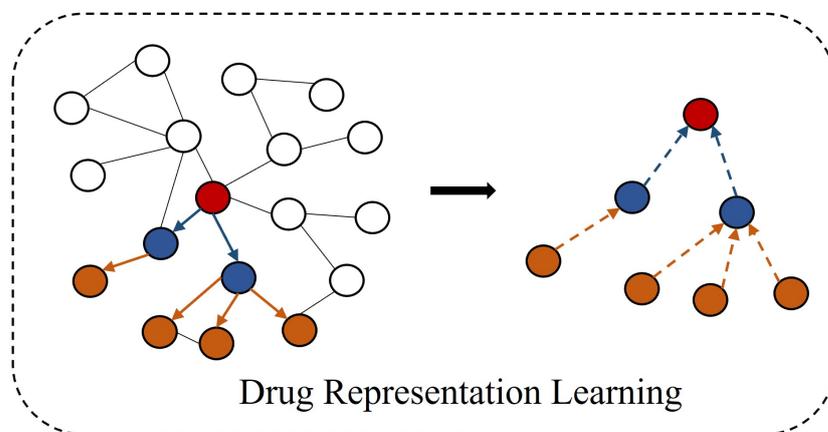
Moreover, we proceed to construct the representation layer by superimposing more propagation layers to lengthen the network paths to learn the global representation of drug nodes. Subsequently, we aggregate the first-order neighboring information propagated from neighbors, as depicted in Figure 3. More specifically, assuming that there are a total of  $l$  propagation layers, the representation of drug node  $v_i$  at the  $l$ th layer is recursively formulated as:

$$h_i^{(l)} = MLP^{(l)}((1 + \epsilon^{(l)})h_i^{(l-1)} + h_{N_i}^{(l-1)}) \tag{6}$$

where  $h_i^{(0)} = h_i$ , and the amount of neighboring information for  $v_i$  in the  $l$ th is defined as:

$$h_{N_i}^{(l-1)} = \sum_{j \in N_i} h_j^{(l-1)} \tag{7}$$

where  $h_j^{(l-1)}$  is the representation of  $v_j$  generated from the previous layer.



**Figure 3.** An illustration for the representation learning layer.

### 2.2.3. DDI Prediction

For a query pair of drugs, i.e.,  $(v_i, v_j)$ , their global representations, i.e.,  $h_i^{(l)}$  and  $h_j^{(l)}$ , can be obtained with (6). Then, we design the scoring function  $\mathcal{F}$  to calculate the interaction probability between  $v_i$  and  $v_j$ , and the definition of  $\mathcal{F}$  is given as:

$$\mathcal{F}(h_i^{(l)}, h_j^{(l)}) = \sigma(h_i^{(l)} \cdot h_j^{(l)}) \tag{8}$$

where  $\sigma$  is defined as the sigmoid( $\cdot$ ) activation function [40] widely adopted to address binary classification problems [41,42].

In our work, we approach the problem of DDI prediction as a binary classification task. Hence, given  $(v_i, v_j)$ , we train DDIGIN by using the binary cross-entropy loss function defined as:

$$\mathcal{L}(\Theta) = - \sum_{i \neq j} p_{(i,j)} \log(\hat{p}_{(i,j)}) + (1 - p_{(i,j)}) \log(1 - \hat{p}_{(i,j)}) \quad (9)$$

where  $\hat{p}_{(i,j)} = \mathcal{F}(h_i^{(l)}, h_j^{(l)})$ ,  $p_{(i,j)}$  represents the binary value indicating the existence of a DDI between  $v_i$  and  $v_j$ , and  $\Theta$  denotes the set of trainable parameters.

### 3. Experiments

#### 3.1. Datasets

For performance evaluation, extensive experiments have been conducted on two benchmark datasets, i.e., the Open Graph Benchmark Drug-Drug Interaction (ogbl-ddi) dataset [43], and the DDInter dataset [44]. These two datasets are different in size, and their statistics are shown in Table 1.

- The ogbl-ddi dataset is an unweighted, undirected graph, where each node represents a Food and Drug Administration (FDA)-approved or experimental drug from the DrugBank 5.0 database [45], and each edge represents interactions between corresponding drugs [43].
- The DDInter dataset is a comprehensive and practical DDI database that currently contains 1833 approved drugs that have been reviewed and curated by a clinical pharmacist team, and approximately 0.23 million DDI pairs [44].

**Table 1.** Statistics of the benchmark datasets used in the experiments.

Dataset	# Nodes	# Edges	# Density
ogbl-ddi	4267	1,334,889	0.1467%
DDInter	1833	222,384	0.1324%

Density is defined as  $\frac{2 * Edges}{Nodes * Nodes}$ .

#### 3.2. Baseline Algorithms

For the purpose of demonstrating the effectiveness of DDIGIN, several classical and state-of-art prediction algorithms are taken as the baseline models, and their brief descriptions are presented as follows.

- LINE [19]: It is an NN-based approach for network representation learning that learns the final representation by designing two kinds of proximities and optimizing them simultaneously.
- SDNE [22]: It can be considered as an extension of LINE, as well as the pioneering method of applying deep learning in graph representation learning through the utilization of an autoencoder.
- GraphSAGE [33]: GraphSAGE is a versatile inductive framework that utilizes node feature information to generate node embeddings effectively, even for data that have not been seen during the training phase.
- GCN [31]: Based on a first-order approximation of spectral convolutions on graphs, it employs an effective layerwise propagation method.
- DPDDI [46]: It extracts the network structure features of drugs from a DDI network with GCN [31,47], and then a deep neural network is trained to predict potential interactions.
- GCN-DTI [48]: It learns drug representations using a traditional GCN model [49] and then adopts a deep neural network to predict the final labels for drug pairs.
- DeepDDS [50]: It learns drug embedding vectors using a graph attention network or GCN, adopts an MLP to extract the cell line features, and then concatenates them to predict the synergy of drug combinations.

- CPI-IGAE [51]: It uses the optimized inductive aggregator based on GraphSAGE for feature extraction, and then designs the scoring function based on the inner product to adumbrate the drug interaction probability.

### 3.3. Experimental Settings

DDIGIN was implemented with PyTorch [52] on a working machine equipped with an Intel Core I7 3.2 GHz and 16 GB of RAM. The above baseline algorithms were also deployed on the same machine. We used a dataset split function provided by OGB for the ogbl-ddi dataset, and we randomly divided the whole dataset into a training set, validation set, and test set with a ratio of 8:1:1 for the DDInter dataset. Moreover, for all baseline algorithms, the embedding size was fixed at 256 and other parameters were consistent with the original work. As for the parameters of DDIGIN, we set the hidden size, learning rate, dropout, and the number of layers of DDIGIN to 256, 0.005, 0.5, and 3, respectively.

### 3.4. Evaluation Metrics

We evaluated the prediction performance of DDIGIN using several classification metrics, including accuracy (Acc), precision, recall, and F1 score. These metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (10)$$

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (13)$$

where TP, FP, TN, and FN represent the numbers of true positive samples, false positive samples, true negative samples, and false negative samples, respectively.

We also use the area under the ROC curve (AUC) and area under the precision–recall curve (AUPR) metrics to measure the performance of DDIGIN from a global perspective without specifying prediction thresholds.

### 3.5. Results

In this section, we evaluate all baselines with the above six evaluation metrics by quantitatively measuring their performance on two benchmark datasets, respectively. In Table 2, we present the results of all the models, where several things can be noted.

First of all, LINE and SDNE are network embedding (NE)-based algorithms, and they solve DDI problems through conventional network representation learning methods, such as MF and RW. Concerning the outcomes achieved by NE-based algorithms, we find that the performance of SDNE is better than LINE. The reason is twofold. On the one hand, LINE adopts a shallow model, while SDNE adopts a deep neural network structure which can capture highly nonlinear network structures effectively [53]. On the other hand, the node representations of both LINE and SDNE are learned by designing first-order and second-order proximities [54]. However, the two optimization methods are different. LINE is optimized separately, while SDNE is optimized simultaneously which can preserve both global and local structures. According to Table 2, it is evident that DDIGIN consistently outperforms NE-based algorithms for each metric across all benchmark datasets. This can be attributed to the fact that the performance of NE-based algorithms is limited by the representation capabilities of corresponding models. DDIGIN learns more robust initial embeddings of drugs with Node2Vec. On average, the performance of DDIGIN is better by 15% than NE-based algorithms on the DDInter dataset across all metrics.

**Table 2.** Experimental results on benchmark datasets.

Datasets	Methods	Acc	F1 Score			AUC	AUPR
			Precision	Recall	F1 Score		
ogbl-ddi	LINE	0.9038	0.9089	0.9033	0.9061	0.9052	0.9147
	SDNE	0.9107	0.9201	0.9176	0.9188	0.9113	0.9228
	GCN	0.9632	0.9692	0.9693	0.9692	0.9642	0.9782
	GraphSAGE	0.9715	0.9712	0.9619	0.9665	0.9717	0.9833
	DPDDI	0.9717	0.9742	0.9640	0.9691	0.9721	0.9836
	GCN-DTI	0.9737	0.9775	0.9649	0.9712	0.9744	0.9848
	DeepDDS	0.9717	0.9749	0.9645	0.9697	0.9721	0.9831
	CPI-IGAE	0.9722	0.9761	0.9651	0.9706	0.9731	0.9842
	GIN	0.9742	0.9792	0.9654	0.9717	0.9762	0.9850
	<b>DDIGIN</b>	<b>0.9763</b>	<b>0.9866</b>	<b>0.9682</b>	<b>0.9773</b>	<b>0.9772</b>	<b>0.9868</b>
DDInter	LINE	0.6604	0.7783	0.6229	0.6920	0.6742	0.8093
	SDNE	0.6631	0.7825	0.6377	0.7027	0.6809	0.7912
	GCN	0.7740	0.7966	0.7993	0.7979	0.7934	0.9170
	GraphSAGE	0.6602	0.6654	0.6534	0.6593	0.7191	0.8868
	DPDDI	0.7903	0.8388	0.8233	0.8310	0.8118	0.9179
	GCN-DTI	0.8071	0.8494	0.8242	0.8366	0.8189	0.9197
	DeepDDS	0.8004	0.8425	0.8237	0.8330	0.8155	0.9193
	CPI-IGAE	0.8066	0.8467	0.8240	0.8352	0.8158	0.9195
	GIN	0.8121	0.8579	0.8255	0.8414	0.8217	0.9287
	<b>DDIGIN</b>	<b>0.8518</b>	<b>0.9372</b>	<b>0.8433</b>	<b>0.8878</b>	<b>0.8594</b>	<b>0.9402</b>

The best results are bolded.

Second, GNN-based algorithms, such as GCN and GraphSAGE, use variations of GNNs to predict DDIs. According to Table 2, it is apparent that DDIGIN exhibits superior performance compared to GNN-based algorithms on all datasets. In addition, it can be observed that the experimental results achieved by DDIGIN are about 1.68% and 11% better on the ogbl-ddi and DDInter datasets, respectively, than GNN-based algorithms. The main reason for this phenomenon is that GNN-based algorithms fail to effectively distinguish some simple graph structures, as they follow the neighborhood aggregation architecture to generate node embedding by recursively aggregating and transforming the feature vectors of neighborhood nodes. However, DDIGIN iteratively updates a given drug node by aggregating feature vectors of its neighbors by using injective aggregation update to map different feature vectors. In doing so, the drug embeddings generated by DDIGIN retain the graph structure identification information. As a result, DDIGIN is able to achieve a better performance when compared with GNN-based algorithms.

Furthermore, we also note that little difference is observed in the performance between GCN and GraphSAGE on ogbl-ddi, while GCN yields a better performance than GraphSAGE on DDInter. Considering that GraphSAGE samples from neighbors of each node during the graph propagation, especially for graphs with unbalanced distribution, its robustness may be poor. Furthermore, it is worth noting that the experimental results of GCN on the DDInter dataset are approximately 14% better than GraphSAGE.

Overall, DDIGIN yields the best performance in terms of all evaluation metrics on two benchmark datasets, and the reasons are twofold. On one hand, DDIGIN incorporates the Node2Vec approach to initialize drug embeddings, thus increasing the flexibility of neighborhood search. On the other hand, the use of GIN allows DDIGIN to generate higher-quality, more information-embedded representations by addressing the problem of isomorphic structures.

### 3.6. Ablation Study

In order to study the influence of embedding initialization and normalization, an ablation study was conducted. In addition, additional experiments were conducted on the

performance of DDIGIN to analyze the effects of various aggregation functions. In the experiment, three variants of DDIGIN are defined as below. Table 3 displays the experimental results of these variants.

- DDIGIN+R: It simply replaces Node2Vec with a random embedding initialization.
- DDIGIN-BN: It simply removes the normalization component.
- DDIGIN<sub>max</sub>: This variant utilizes an alternative aggregation function, namely,  $\mathcal{A}_{max}$ , to aggregate the information in  $\mathcal{A}_{max} = \sigma(W_{max} \cdot MAX(\mathbf{h}_i^{(l)}, \mathbf{h}_{N_i}^{(l-1)}))$  by following [33].
- DDIGIN<sub>mean</sub>: A modified aggregation function, i.e.,  $\mathcal{A}_{mean}$ , is employed in this variant to aggregates the information in  $\mathcal{A}_{mean} = \sigma(W_{mean} \cdot MEAN(\mathbf{h}_i^{(l)}, \mathbf{h}_{N_i}^{(l-1)}))$  by following [31].
- DDIGIN<sub>sum</sub>: The difference between DDIGIN<sub>sum</sub> and DDIGIN is that DDIGIN<sub>sum</sub> uses  $\mathcal{A}_{sum}$  as its aggregation function, which is defined as  $\mathcal{A}_{sum} = \sigma(W_{sum} \cdot (\mathbf{h}_i^{(l)} + \mathbf{h}_{N_i}^{(l-1)}))$  by following [31].

**Table 3.** Experimental results of the ablation study on the ogbl-ddi and DDInter datasets.

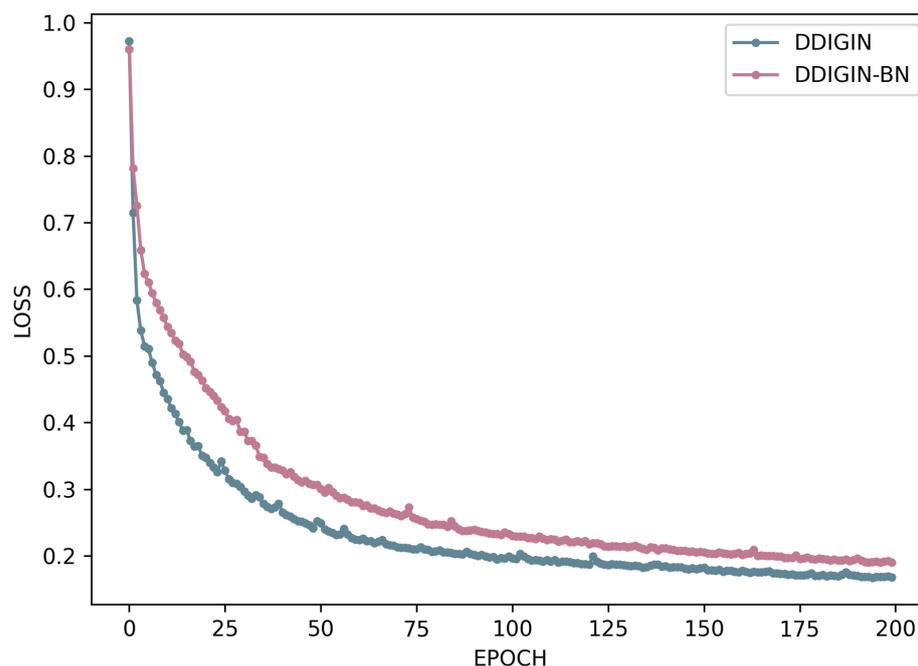
Datasets	Methods	Acc	F1 Score			AUC	AUPR
			Precision	Recall	F1 Score		
ogbl-ddi	DDIGIN+R	0.9679	0.9792	0.9643	0.9717	0.9673	0.9821
	DDIGIN-BN	0.9718	0.9820	0.9793	0.9806	0.9723	0.9836
	DDIGIN <sub>max</sub>	0.9635	0.9727	0.9648	0.9687	0.9637	0.9790
	DDIGIN <sub>mean</sub>	0.9692	0.9803	0.9652	0.9727	0.9700	0.9836
	DDIGIN <sub>sum</sub>	0.9718	0.9851	0.9673	0.9761	0.9724	0.9855
	<b>DDIGIN</b>	<b>0.9763</b>	<b>0.9866</b>	<b>0.9682</b>	<b>0.9773</b>	<b>0.9772</b>	<b>0.9868</b>
DDInter	DDIGIN+R	0.8169	0.9160	0.8104	0.8600	0.8199	0.9273
	DDIGIN-BN	0.8083	0.9168	0.8034	0.8564	0.8186	0.9272
	DDIGIN <sub>max</sub>	0.7669	0.9062	0.7624	0.8281	0.7909	0.9164
	DDIGIN <sub>mean</sub>	0.8263	0.9254	0.8234	0.8714	0.8306	0.9320
	DDIGIN <sub>sum</sub>	0.8495	0.9260	0.8471	0.8848	0.8518	0.9396
	<b>DDIGIN</b>	<b>0.8518</b>	<b>0.9372</b>	<b>0.8433</b>	<b>0.8878</b>	<b>0.8594</b>	<b>0.9402</b>

The best results are bolded.

In the aggregation functions of DDIGIN<sub>max</sub>, DDIGIN<sub>mean</sub>, and DDIGIN<sub>sum</sub>,  $W$  represents the trainable parameters. For DDIGIN, it should be noted that DDIGIN adopts  $\mathcal{A}_{sum} = \sigma(MLP^{(l)}((1 + \epsilon^{(l)})\mathbf{h}_i^{(l-1)} + \mathbf{h}_{N_i}^{(l-1)}))$ .

**Effect of drug embedding initialization.** According to previous studies [55,56], Node2Vec outperforms other existing state-of-the-art methods in node embedding. As can be seen from the results in Table 3, when Node2Vec is replaced by a random initialization, the performance of DDIGIN on some metrics decreases to varying degrees, thus indicating the importance of Node2Vec. A possible reason for this is that Node2Vec is a biased random walk algorithm, and it adjusts different walk situations through hyperparameters to make it explore more homogeneously or structurally, and thereby improves the performance of DDIGIN considerably.

**Effect of BatchNorm.** Comparing the experimental results of DDIGIN and DDIGIN-BN, it is evident that DDIGIN exhibits superior performance on some metrics when employing the normalization strategy which normalizes each layer to act as a decoupling. We can see from Figure 4 that BatchNorm can accelerate the convergence rate of DDIGIN and prevent overfitting.



**Figure 4.** LOSS curves obtained by DDIGIN and DDIGIN-BN.

**Effect of the aggregation function.** To evaluate the effectiveness of aggregation functions, the performance of DDIGIN was compared with three variants, i.e.,  $DDIGIN_{max}$ ,  $DDIGIN_{mean}$ , and  $DDIGIN_{sum}$ . The results presented in Table 3 yield several observations. Firstly, updating node embedding only by their max embeddings is not able to learn more distinguishable characteristics, as  $DDIGIN_{max}$  achieves the worst performance across all datasets. Second, compared to  $DDIGIN_{max}$ ,  $DDIGIN_{mean}$  performs a little better. Last,  $DDIGIN_{sum}$  is better than the other aggregation functions, but it is not an optimal solution, as it simply treats  $h_i^{(l)}$  and  $h_{N_i}^{(l-1)}$  without any difference, while DDIGIN uses a more rational aggregation function by enhancing the relationship between them.

### 3.7. Case Study

#### 3.7.1. Predicting Novel DDIs

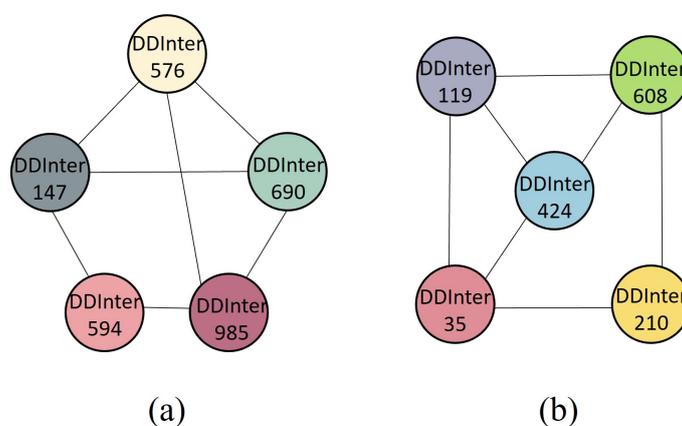
We investigated the performance of DDIGIN in predicting unobserved DDIs. Out of a total of 1833 drugs, DDInter contains 175,202 pairs of observed drug–drug interactions and 47,182 pairs of unobserved DDIs. By training DDIGIN with the known DDI network from the DDInter dataset as training samples, the possible interactions among drugs were inferred. If an unknown drug pair is predicted with higher scores, they are more likely to interact with each other. We collected the top 15 DDIs predicted by DDIGIN and present them in Table 4. By searching for the evidence of these newly predicted DDIs in the DrugBank database [45], we found that nine were confirmed. This result is a strong indicator that DDIGIN demonstrates a superior capability in predicting potential DDIs. Therefore, it can be considered a promising tool for gaining new insights into the prediction of novel DDIs by utilizing a GIN to obtain feature representations.

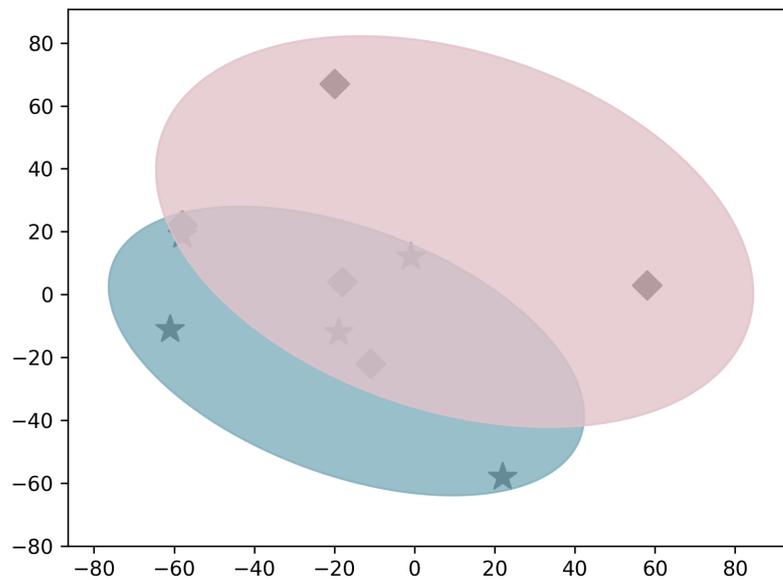
**Table 4.** Top 15 potential DDIs predicted by DDIGIN.

Number	Drug	Drug	Evidence
1	Metformin	Tacrolimus	The therapeutic effectiveness of metformin may be diminished when metformin is used in conjunction with tacrolimus.
2	Prednisolone	Fexofenadine	N/A
3	Nystatin	Metronidazole	N/A
4	Epinephrine	Salbutamol	The combination of epinephrine and salbutamol can increase the risk or severity of adverse effects.
5	Cetirizine	Prednisolone	N/A
6	Leflunomide	Dexamethasone	When dexamethasone is combined with leflunomide, the risk or severity of adverse effects can be heightened.
7	Tamsulosin	Promethazine	The metabolism of tamsulosin can be decreased when combined with promethazine.
8	Zolpidem	Nystatin	N/A
9	Nystatin	Quetiapine	N/A
10	Valsartan	Nystatin	The excretion of valsartan can be decreased when combined with nystatin.
11	Triamcinolone	Fentanyl	The metabolism of fentanyl can be increased when combined with triamcinolone.
12	Nabumetone	Prednisolone	When prednisolone is combined with nabumetone, there is an increased risk or severity of gastrointestinal irritation.
13	Prednisolone	Insulin degludec	When prednisolone is combined with insulin degludec, there is an elevated risk or severity of hyperglycemia.
14	Folic acid	Furosemide	The combination of furosemide and folic acid may lead to an increased excretion rate of folic acid, potentially resulting in lower serum levels and a potential reduction in efficacy.
15	Lansoprazole	Prednisone	N/A

### 3.7.2. Distinguishing Similar Structures

To test the model's ability to distinguish similar structures, we selected two subgraphs with a similar structure from the DDInter dataset, as shown in Figure 5. Moreover, the learned embeddings of drugs in these two subgraphs are visualized in Figure 6. Traditional models attempt to learn similar node embedding for drugs by following similar aggregation paths. However, we can see from Figure 6 that DDIGIN generates different embeddings for drugs in these two subgraphs. The reason is that the most important feature of DDIGIN in the aggregation process is that its aggregation function uses injective functions to distinguish between different graph structures. As a result, the representations learned for drugs by DDIGIN are more discriminative even for those within isomorphic structures, thus improving the performance of DDI prediction.

**Figure 5.** An example of graph isomorphism problem between (a) and (b) on the DDInter dataset.

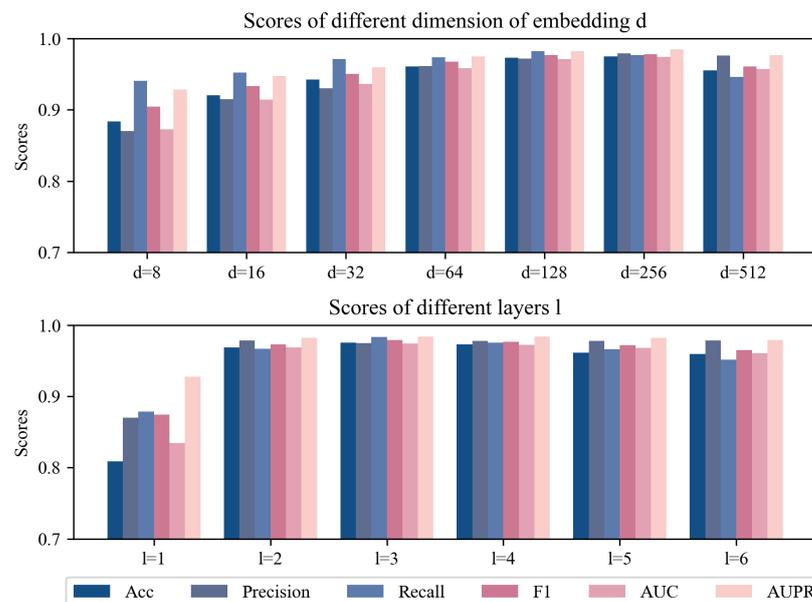


**Figure 6.** The visualization results of two subgraphs. The blue area represents the visualization range of Figure 5a, and the pink area represents the visualization range of Figure 5b.

### 3.8. Parameter Sensitivity Analysis

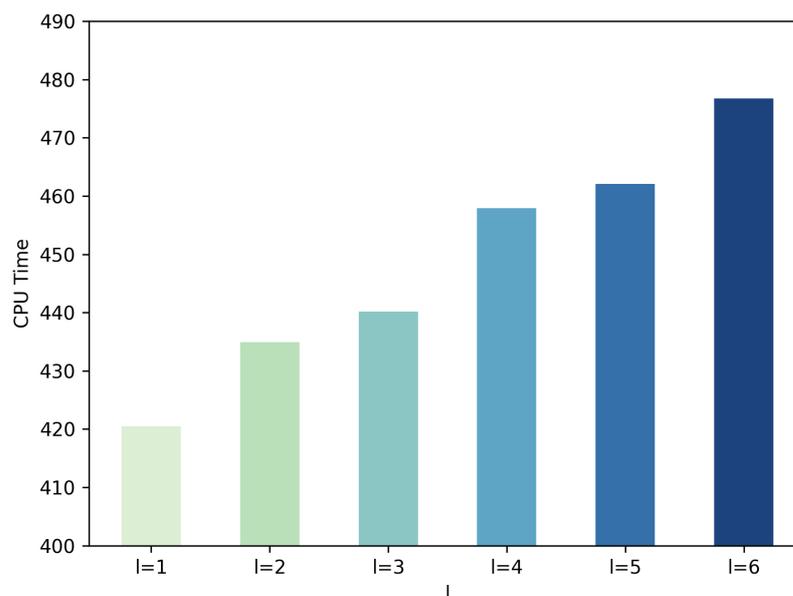
A parameter sensitivity study was further conducted to evaluate the performance of DDIGIN with different parameter settings. In particular, we investigated the effects of two hyperparameters that needed to be tuned, the dimension of drug embeddings, i.e.,  $d$ , and the number of layers, i.e.,  $l$ . Taking the ogbl-ddi dataset as an example, the performance of DDIGIN in terms of different evaluation metrics is presented in Figure 7.

**Effect of dimension of embedding.** We examined the influence of  $d$  by varying its value from 8 to 512. The result is rather intuitive, as can be seen from Figure 7, where with the increase in the embedding dimension  $d$ , each evaluation metric increases to varying degrees. The best effect is achieved when the dimension is set as 128. A further increase in  $d$  degrades the performance of DDIGIN to some extent, which may be caused by overfitting.



**Figure 7.** Results of DDIGIN with a varying size of  $d$  and  $l$ .

**Effect of the number of layers of DDIGIN.** We investigated the influence of the number of layers of DDIGIN by varying the value of  $l$  from one to six. According to Figure 7, it can be observed that DDIGIN obtains its best performance when  $l$  is set to two or three. As  $l$  increases, the number of learning nodes for the representation of a particular drug node increases. Therefore, DDIGIN is prone to encounter noise data due to the increase in  $l$ . Consequently, the performance of DDIGIN degrades when  $l > 3$ . Additionally, we also note in Figure 8 that DDIGIN trained with a larger  $l$  requires more CPU time to reach convergence due to the increase in involved nodes. Thus,  $l = 3$  is often sufficient for the actual situation of DDI prediction.



**Figure 8.** The change in CPU time taken by DDIGIN given different values of  $l$ .

#### 4. Conclusions

In this work, DDIGIN was proposed to identify potential DDIs. First, DDIGIN initialized the representations of drugs with Node2Vec according to the topological structure and then optimized these representations by propagating and aggregating the first-order neighboring information in an injective way. Last, it determined the interaction probability for pairwise drugs by multiplying their representations in an end-to-end manner. Experimental results showed that DDIGIN outperformed several state-of-the-art algorithms on the DDI prediction task when using an injective aggregation function, and the incorporation of GIN enhanced the expressive power of drug representations for an improved performance of DDI prediction.

With regard to future work, there is room for further improvement in the performance of DDIGIN. Due to the ubiquity of the knowledge graph, we can use a KG to predict DDI and extract drug characteristics [34,57], which provides more detailed information about drug attributes and drug-related triple facts. In addition, there are other types of associations in bioinformatics, such as protein–protein interactions [58–61] and drug–disease associations [62], and we may also employ DDIGIN to predict them. However, deep learning models have a significant interpretability problem, which can negatively impact the model’s performance. We recommend that studies in the future think about developing more comprehensible models.

**Author Contributions:** S.W., X.S. and L.H. conceived the main idea and the framework of the manuscript and performed the experiments; T.B., B.Z. and P.H. analyzed the results. X.S. and L.H. helped to improve the idea and the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been supported by the Xinjiang Uygur Autonomous region universities basic research business funds research projects under grant XJEDU2022J009, Natural Science Foundation of Xinjiang Uygur Autonomous Region under grant 2021D01D05, Tianshan Youth Project–Outstanding Youth Science and Technology Talents of Xinjiang under grant 2020Q005, CAS Light of the West Multidisciplinary Team project under grant xbzg-zdsys-202114, the Pioneer Hundred Talents Program of Chinese Academy of Sciences, the National Key R&D Program of China under grant 2022ZD0115805, and the Provincial Key S&T Program of Xinjiang under grant 2022A02011.

**Data Availability Statement:** The ogbl-ddi datasets are available at <https://ogb.stanford.edu/docs/linkprop/#ogbl-ddi>, accessed on 15 June 2023, and the DDInter dataset is available at <http://ddinter.scbdd.com/>, accessed on 23 June 2023. The dataset and source code during the current study can be freely downloaded from <https://github.com/WSL0410/DDIGIN>, accessed on 1 September 2023.

**Acknowledgments:** The authors would like to thank colleagues and the anonymous reviewers for the valuable comments on the original manuscript.

**Conflicts of Interest:** The authors declare no conflicts of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

DDI	Drug–drug interaction
GIN	Graph isomorphism network
GNN	Graph neural networks
GCN	Graph convolutional network
KG	Knowledge graph

### References

1. Giacomini, K.M.; Krauss, R.M.; Roden, D.M.; Eichelbaum, M.; Hayden, M.R.; Nakamura, Y. When good drugs go bad. *Nature* **2007**, *446*, 975–977. [CrossRef] [PubMed]
2. Magro, L.; Moretti, U.; Leone, R. Epidemiology and characteristics of adverse drug reactions caused by drug–drug interactions. *Expert Opin. Drug Saf.* **2012**, *11*, 83–94. [CrossRef] [PubMed]
3. Lazarou, J.; Pomeranz, B.H.; Corey, P.N. Incidence of adverse drug reactions in hospitalized patients: A meta-analysis of prospective studies. *JAMA* **1998**, *279*, 1200–1205. [CrossRef] [PubMed]
4. Vilar, S.; Uriarte, E.; Santana, L.; Lorberbaum, T.; Hripcsak, G.; Friedman, C.; Tatonetti, N.P. Similarity-based modeling in large-scale prediction of drug–drug interactions. *Nat. Protoc.* **2014**, *9*, 2147–2163. [CrossRef]
5. Qato, D.M.; Wilder, J.; Schumm, L.P.; Gillet, V.; Alexander, G.C. Changes in prescription and over-the-counter medication and dietary supplement use among older adults in the United States, 2005 vs. 2011. *JAMA Intern. Med.* **2016**, *176*, 473–482. [CrossRef]
6. Percha, B.; Altman, R.B. Informatics confronts drug–drug interactions. *Trends Pharmacol. Sci.* **2013**, *34*, 178–184. [CrossRef]
7. Becker, M.L.; Kallewaard, M.; Caspers, P.W.; Visser, L.E.; Leufkens, H.G.; Stricker, B.H. Hospitalisations and emergency department visits due to drug–drug interactions: A literature review. *Pharmacoepidemiol. Drug Saf.* **2007**, *16*, 641–651. [CrossRef]
8. Lin, X.; Quan, Z.; Wang, Z.J.; Huang, H.; Zeng, X. A novel molecular representation with BiGRU neural networks for learning atom. *Briefings Bioinform.* **2020**, *21*, 2099–2111. [CrossRef]
9. Zhang, D.; Yin, J.; Zhu, X.; Zhang, C. Network representation learning: A survey. *IEEE Trans. Big Data* **2018**, *6*, 3–28. [CrossRef]
10. Zhang, Y.; Qiu, Y.; Cui, Y.; Liu, S.; Zhang, W. Predicting drug–drug interactions using multi-modal deep auto-encoders based network embedding and positive-unlabeled learning. *Methods* **2020**, *179*, 37–46. [CrossRef]
11. Zhao, B.W.; Wang, L.; Hu, P.W.; Wong, L.; Su, X.R.; Wang, B.Q.; You, Z.H.; Hu, L. Fusing higher and lower-order biological information for drug repositioning via graph representation learning. *IEEE Trans. Emerg. Top. Comput.* **2023**. [CrossRef]
12. Su, X.; You, Z.; Wang, L.; Hu, L.; Wong, L.; Ji, B.; Zhao, B. SANE: A sequence combined attentive network embedding model for COVID-19 drug repositioning. *Appl. Soft Comput.* **2021**, *111*, 107831. [CrossRef]
13. Belkin, M.; Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **2003**, *15*, 1373–1396. [CrossRef]
14. Cao, S.; Lu, W.; Xu, Q. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; pp. 891–900.
15. Zhang, W.; Chen, Y.; Li, D.; Yue, X. Manifold regularized matrix factorization for drug–drug interaction prediction. *J. Biomed. Inform.* **2018**, *88*, 90–97. [CrossRef] [PubMed]
16. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.

17. Zhang, S.; Huang, Z.; Zhou, H.; Zhou, Z. Sce: Scalable network embedding from sparsest cut. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 23–27 August 2020; pp. 257–265.
18. Purkayastha, S.; Mondal, I.; Sarkar, S.; Goyal, P.; Pillai, J.K. Drug-drug interactions prediction based on drug embedding and graph auto-encoder. In Proceedings of the 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, 28–30 October 2019; pp. 547–552.
19. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
20. Dong, Y.; Chawla, N.V.; Swami, A. metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 135–144.
21. Liu, S.; Zhang, Y.; Cui, Y.; Qiu, Y.; Deng, Y.; Zhang, Z.M.; Zhang, W. Enhancing drug-drug interaction prediction using deep attention neural networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2022**, *20*, 976–985. [[CrossRef](#)] [[PubMed](#)]
22. Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234.
23. Hu, L.; Pan, X.; Yan, H.; Hu, P.; He, T. Exploiting higher-order patterns for community detection in attributed graphs. *Integr. Comput. Aided Eng.* **2021**, *28*, 207–218. [[CrossRef](#)]
24. Hu, L.; Yang, Y.; Tang, Z.; He, Y.; Luo, X. FCAN-MOPSO: An Improved Fuzzy-based Graph Clustering Algorithm for Complex Networks with Multi-objective Particle Swarm Optimization. *IEEE Trans. Fuzzy Syst.* **2023**. [[CrossRef](#)]
25. Feng, Y.H.; Zhang, S.W. Prediction of drug-drug interaction using an attention-based graph neural network on drug molecular graphs. *Molecules* **2022**, *27*, 3004. [[CrossRef](#)]
26. Bai, Y.; Gu, K.; Sun, Y.; Wang, W. Bi-level graph neural networks for drug-drug interaction prediction. *arXiv* **2020**, arXiv:2006.14002.
27. Chen, X.; Liu, X.; Wu, J. GCN-BMP: Investigating graph representation learning for DDI prediction task. *Methods* **2020**, *179*, 47–54. [[CrossRef](#)]
28. Ma, M.; Lei, X. A dual graph neural network for drug–drug interactions prediction based on molecular structure and interactions. *PLoS Comput. Biol.* **2023**, *19*, e1010812. [[CrossRef](#)]
29. Zitnik, M.; Agrawal, M.; Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **2018**, *34*, i457–i466. [[CrossRef](#)] [[PubMed](#)]
30. Asada, M.; Miwa, M.; Sasaki, Y. Enhancing drug-drug interaction extraction from texts by molecular structure information. *arXiv* **2018**, arXiv:1805.05593.
31. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
32. Zhou, H.; Zhou, W.; Wu, J. Graph Distance Neural Networks for Predicting Multiple Drug Interactions. *arXiv* **2022**, arXiv:2208.14810.
33. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1025–1035.
34. Su, X.; Hu, L.; You, Z.; Hu, P.; Zhao, B. Attention-based knowledge graph representation learning for predicting drug-drug interactions. *Briefings Bioinform.* **2022**, *23*, bbac140. [[CrossRef](#)]
35. Leman, A.; Weisfeiler, B. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Tech. Informatsiya* **1968**, *2*, 12–16.
36. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Deep graph contrastive representation learning. *arXiv* **2020**, arXiv:2006.04131.
37. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
38. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
39. Makarov, I.; Savchenko, A.; Korovko, A.; Sherstyuk, L.; Severin, N.; Mikheev, A.; Babaev, D. Temporal graph network embedding with causal anonymous walks representations. *arXiv* **2021**, arXiv:2108.08754.
40. Han, J.; Moraga, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In Proceedings of the From Natural to Artificial Neural Computation: International Workshop on Artificial Neural Networks Malaga-Torremolinos, Malaga-Torremolinos, Spain, 7–9 June 1995; Proceedings 3; Springer: Berlin/Heidelberg, Germany, 1995; pp. 195–201.
41. Daqi, G.; Yan, J. Classification methodologies of multilayer perceptrons with sigmoid activation functions. *Pattern Recognit.* **2005**, *38*, 1469–1482. [[CrossRef](#)]
42. Sharma, R.; Shrivastava, S.; Kumar Singh, S.; Kumar, A.; Saxena, S.; Kumar Singh, R. AniAMPpred: Artificial intelligence guided discovery of novel antimicrobial peptides in animal kingdom. *Briefings Bioinform.* **2021**, *22*, bbab242. [[CrossRef](#)] [[PubMed](#)]
43. Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 22118–22133.
44. Xiong, G.; Yang, Z.; Yi, J.; Wang, N.; Wang, L.; Zhu, H.; Wu, C.; Lu, A.; Chen, X.; Liu, S.; et al. DDInter: An online drug–drug interaction database towards improving clinical decision-making and patient safety. *Nucleic Acids Res.* **2022**, *50*, D1200–D1207. [[CrossRef](#)]

45. Wishart, D.S.; Feunang, Y.D.; Guo, A.C.; Lo, E.J.; Marcu, A.; Grant, J.R.; Sajed, T.; Johnson, D.; Li, C.; Sayeeda, Z.; et al. DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Res.* **2018**, *46*, D1074–D1082. [[CrossRef](#)]
46. Feng, Y.H.; Zhang, S.W.; Shi, J.Y. DPDDI: A deep predictor for drug–drug interactions. *BMC Bioinform.* **2020**, *21*, 1–15. [[CrossRef](#)]
47. Kipf, T.N.; Welling, M. Variational graph auto-encoders. *arXiv* **2016**, arXiv:1611.07308.
48. Zhao, T.; Hu, Y.; Valsdottir, L.R.; Zang, T.; Peng, J. Identifying drug–target interactions based on graph convolutional network and deep neural network. *Briefings Bioinform.* **2021**, *22*, 2141–2150. [[CrossRef](#)]
49. Gao, H.; Wang, Z.; Ji, S. Large-scale learnable graph convolutional networks. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1416–1424.
50. Wang, J.; Liu, X.; Shen, S.; Deng, L.; Liu, H. DeepDDS: Deep graph neural network with attention mechanism to predict synergistic drug combinations. *Briefings Bioinform.* **2022**, *23*, bbab390. [[CrossRef](#)]
51. Wan, X.; Wu, X.; Wang, D.; Tan, X.; Liu, X.; Fu, Z.; Jiang, H.; Zheng, M.; Li, X. An inductive graph neural network model for compound–protein interaction prediction based on a homogeneous graph. *Briefings Bioinform.* **2022**, *23*, bbac073. [[CrossRef](#)]
52. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
53. Hong, R.; He, Y.; Wu, L.; Ge, Y.; Wu, X. Deep attributed network embedding by preserving structure and attribute information. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 1434–1445. [[CrossRef](#)]
54. Xie, Y.; Gong, M.; Wang, S.; Liu, W.; Yu, B. Sim2vec: Node similarity preserving network embedding. *Inf. Sci.* **2019**, *495*, 37–51. [[CrossRef](#)]
55. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowl.-Based Syst.* **2018**, *151*, 78–94. [[CrossRef](#)]
56. Yue, X.; Wang, Z.; Huang, J.; Parthasarathy, S.; Moosavinasab, S.; Huang, Y.; Lin, S.M.; Zhang, W.; Zhang, P.; Sun, H. Graph embedding on biomedical networks: Methods, applications and evaluations. *Bioinformatics* **2020**, *36*, 1241–1251. [[CrossRef](#)] [[PubMed](#)]
57. Lin, X.; Quan, Z.; Wang, Z.J.; Ma, T.; Zeng, X. KGNN: Knowledge Graph Neural Network for Drug-Drug Interaction Prediction. In Proceedings of the IJCAI, Yokohama, Japan, 11–17 July 2020; Volume 380, pp. 2739–2745.
58. Hu, L.; Wang, X.; Huang, Y.A.; Hu, P.; You, Z.H. A novel network-based algorithm for predicting protein–protein interactions using gene ontology. *Front. Microbiol.* **2021**, *12*, 735329. [[CrossRef](#)]
59. Hu, L.; Wang, X.; Huang, Y.A.; Hu, P.; You, Z.H. A survey on computational models for predicting protein–protein interactions. *Briefings Bioinform.* **2021**, *22*, bbab036. [[CrossRef](#)]
60. Hu, L.; Chan, K.C. Discovering variable-length patterns in protein sequences for protein–protein interaction prediction. *IEEE Trans. Nanobiosci.* **2015**, *14*, 409–416. [[CrossRef](#)]
61. Hu, L.; Zhang, J.; Pan, X.; Luo, X.; Yuan, H. An effective link-based clustering algorithm for detecting overlapping protein complexes in protein–protein interaction networks. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 3275–3289. [[CrossRef](#)]
62. Zhao, B.W.; Su, X.R.; Hu, P.W.; Huang, Y.A.; You, Z.H.; Hu, L. iGRLDTI: An Improved Graph Representation Learning Method for Predicting Drug–Target Interactions over Heterogeneous Biological Information Network. *Bioinformatics* **2023**, *39*, btad451. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.